Pattern Analysis and Applications manuscript No.
(will be inserted by the editor)

A Simple Iterative Algorithm for Parsimonious Binary Kernel Fisher Discrimination

Robert F. Harrison, Kitsuchart Pasupa[⋆]

Department of Automatic Control & Systems Engineering, The University of Sheffield, Sheffield, S1 3JD, England.

Received: date / Revised version: date

Abstract By applying recent results in optimization theory variously known as optimization transfer or majorize/minimize algorithms, an algorithm for binary, kernel, Fisher discriminant analysis is introduced that makes use of a non-smooth penalty on the coefficients to provide a parsimonious solution. The problem is converted into a smooth optimization that can be solved iteratively with no greater overhead than iteratively re-weighted least-squares. The result is simple, easily programmed and is shown to perform, in terms of both accuracy and parsimony, as well as or better than a number of leading machine learning algorithms on two well-studied and substantial benchmarks.

Key words Kernel machines – Fisher discriminant analysis – majorize-minimize algorithms – sparsity – parsimony

1 Introduction

Dimensionality reduction is an important step in pattern recognition and classification where data may exist in high dimensions and Fisher discriminant analysis (FDA) has played a central role in achieving this. FDA seeks a linear projection that maximizes the separation between data belonging to two classes while minimizing the separation between those of the same class. Its properties are well-documented and under certain circumstances prove optimal [1]. However, the linearity of the approach is frequently insufficient to allow the required level of performance in practical applications. While

⋆ *Present address:* School of Electronics & Computer Science, University of Southampton, Southampton, England, SO17 1BJ, UK.
Correspondence to: r.f.harrison@sheffield.ac.uk (Robert F. Harrison)

explicit expansion of data in basis functions can resolve this for problems of low dimension, the combinatorial increase in the number of coefficients to be estimated may make this impractical. Recent focus on kernel machines in the machine learning community seeks to address this problem via the so-called "kernel trick" [2] and a number of solutions have been provided (see e.g. [3–13]) that can be thought of generically as kernel Fisher discriminant analysis (kFDA). While kernels lend the required degree of flexibility to the discrimination task, they bring their own challenges, the foremost being a potential to overspecialize to the sample data and a computational complexity dominated by sample size which, in some problems, may be large. Complexity control is therefore essential for a good outcome yet it has not been widely explored in the context of kFDA. In [4] complexity is controlled through explicit regularization – placing an appropriate penalty on the coefficients of the estimator and solving by mathematical programming, while [14] exploits the connection between FDA and an associated least-squares problem where an orthogonalization technique based on the modified Gram-Schmidt procedure is used for forward regressor selection. In benchmarks, the latter technique is seen to be competitive with a number of leading machine-learning classifiers including kFDA while providing more parsimonious estimators and is used for direct comparison here, along with other results. Another approach that might have application in the kFDA problem is described in [15].

In this paper we again exploit the association of FDA with least-squares and control complexity by penalizing the objective function. It is well-known that penalty functions that induce sparsity lead to non-smooth formulations and these are traditionally solved via mathematical programming techniques as is done in [4]. In a departure, we apply a *majorize-minimize* technique to overcome this technical problem leading to a very simple iterative algorithm that converges to the (penalized) least-squares solution. In [16] a general majorize-minimize framework is presented for variable selection via penalized maximum likelihood but there only a small least-squares problem in conjunction with the SCAD (smoothly-clipped absolute deviations) penalty is examined.

Links between least-squares and FDA solutions are well-known in the binary case [1,17] and [5] makes use of the method propounded in [18] using "optimal scoring"[1] to achieve multinomial discriminant analysis using kernels – the issue of parsimony is not, however, addressed. The extension of the proposed method to more than two classes via this route is more challenging since (except in the trivial, non-parsimonious case), for $C > 2$ classes, the solution generates C operators (corresponding to the "hat" or smoother matrix in conventional regression) and it is not yet clear how these can be related to the multinomial linear discriminant co-ordinates. Of course, one-versus-one and one-versus-remainder strategies can be employed however, since each of these dichotomizers is trained independently, the resulting set may not prove to be parsimonious overall.

[1] Leading to a linear regression followed by a small eigen-decomposition.

The paper is organized as follows. Section 2 briefly states the well-known link between FDA and least-squares [1], presents the kernel-based formulation and motivates the use of penalized optimization. The following section introduces the majorize-minimize principle and sketches a derivation of the iterative algorithm. Section 4 presents a performance comparison with other leading machine learning methods on two well-studied sets of benchmarks.

2 Fisher Discriminant Analysis

The relationship between FDA and least-squares is well known [1]. Consider the matrix of m-dimensional sample vectors $\mathrm{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N]^\mathsf{T}$ comprising two groups, $\mathcal{G}_i$, of size, N_i, $i = 1, 2$ represented by the partition, $\left[\mathrm{U}_1^\mathsf{T}\ \mathrm{U}_2^\mathsf{T}\right]^\mathsf{T}$. Membership of $\mathcal{G}_1$ is denoted by $\hat{y} = +N/N_1$ and of $\mathcal{G}_2$ by $\hat{y} = -N/N_2$ then it is straightforward to verify that the solution for $\left[b\ \boldsymbol{w}^\mathsf{T}\right]^\mathsf{T}$, to the following least-squares problem lies in the same direction as the solution for the Fisher discriminant [1].

$$\arg\min_{(b,\boldsymbol{w})} \left\| \begin{bmatrix} \frac{N}{N_1}\mathbf{1}_{N_1} \\ -\frac{N}{N_2}\mathbf{1}_{N_2} \end{bmatrix} - \begin{bmatrix} \mathbf{1}_{N_1} & \mathrm{U}_1 \\ \mathbf{1}_{N_2} & \mathrm{U}_2 \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{w} \end{bmatrix} \right\|_2^2 \tag{1}$$

where $\mathbf{1}_p$ denotes a p-vector of ones.

To accommodate more complex discriminants, data can be mapped into a new feature space, $\mathcal{F}$, via some function, $\boldsymbol{\phi} : \mathbb{R}^m \mapsto \mathbb{R}^\nu$, say. However, vectors in $\mathcal{F}$ will typically be of very high, or even infinite, dimension, precluding any practical manipulation. The kernel trick recognizes that the coefficients, $\boldsymbol{w}$, in the linear model implicit in (1) can themselves be written as a linear combination of the mapped data, $\boldsymbol{w} = \sum_{i=1}^{i=N} \alpha_i \boldsymbol{\phi}(\boldsymbol{u}_i)$, leading to a formulation entirely based on inner products that can be computed through the agency of a suitable kernel. These ideas have been explored thoroughly elsewhere (see e.g. [2]) so we provide only a skeleton exposition of the kernelized version of the least-squares problem (see e.g. [14] for details).

Briefly, arranging the mapped data into a $\nu \times N$-dimensional matrix, $\boldsymbol{\Phi}$, $\boldsymbol{w}$ can be re-written $\boldsymbol{w} = \boldsymbol{\Phi}\boldsymbol{\alpha}$ and the result as,

$$\left[\mathbf{1}_N\ \boldsymbol{\Phi}^\mathsf{T}\right] \begin{bmatrix} b \\ \boldsymbol{w} \end{bmatrix} = \left[\mathbf{1}_N\ \boldsymbol{\Phi}^\mathsf{T}\right] \begin{bmatrix} b \\ \boldsymbol{\Phi}\boldsymbol{\alpha} \end{bmatrix} = \left[\mathbf{1}_N\ K\right] \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} \tag{2}$$

where $K = \boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi}$ denotes the Gram matrix associated with a suitable kernel, $k(.,.)$, i.e. $k_{ij} = k(\boldsymbol{u}_i, \boldsymbol{u}_j)$, $i, j = 1, 2, \ldots, N$. The solution, $\boldsymbol{\omega} = \left[b\ \boldsymbol{\alpha}^\mathsf{T}\right]^\mathsf{T}$, to the following least-squares problem provides the coefficients of a linear discriminant in the feature space associated with $k(.,.)$, hence a non-linear discriminant in the original space containing the data (see e.g. [4]).

$$\arg\min_{(b,\boldsymbol{\alpha})} \left\| \begin{bmatrix} \frac{N}{N_1}\mathbf{1}_{N_1} \\ -\frac{N}{N_2}\mathbf{1}_{N_2} \end{bmatrix} - \begin{bmatrix} \mathbf{1}_{N_1} & \mathrm{K}_1 \\ \mathbf{1}_{N_2} & \mathrm{K}_2 \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} \right\|_2^2 \tag{3}$$

2.1 Complexity control

It is common to introduce a quadratic or ridge penalty on the coefficients into least-squares regression and this can be interpreted in the Bayesian framework as placing a Gaussian prior on the values of the coefficients, e.g. [19]. In addition to reducing coefficient magnitudes where possible, the quadratic penalty improves numerical condition when data are strongly correlated, militates against over-fitting and also suggests a method for selecting variables – those with relatively small coefficient magnitudes can be discarded. Evidently, such an approach is not optimal but may still lead to adequate performance. The quadratic penalty tends to discourage large values but permits many small values to remain and these may, collectively, contribute substantially to the result. Instead, a penalty corresponding to a prior distribution with a sharp peak has the effect of penalizing non-zero coefficients much more strongly. The pay-off for setting small coefficients exactly to zero instead of just reducing their magnitude is therefore relatively much greater. A penalty of the form $\rho \|\boldsymbol{\omega}\|_q^q$, $0 < q \leq 1$ $\rho \geq 0$, among others, has precisely this property[2]. This "sparsity-inducing" property is well-studied when $q = 1$, for example, as the well-known Lasso estimator in statistics [18,20–22] and has been used widely in the field of kernel machines [2]. A choice of $q < 1$ exacerbates this effect as shown in figure 1. Introducing penalties of this form means that closed-form solutions are no longer possible and leads to difficulties in gradient-based optimization owing to their discontinuous first derivatives. Mathematical programming is often used to address this e.g. in the case that $q = 1$. Here we exploit the majorize-minimize principle to provide a simple, iterative algorithm. The choice of $0 < q < 1$ leads to a further difficulty – the loss of convexity in the penalized objective function so that convergence of the resulting algorithms will be towards a local, rather than global, optimum. This is illustrated in figure 2 for a simple linear regression in one dimension. The lack of convexity is clear for the choice of parameters ($q = 0.25$, $\rho = 3.5$) and two initializations of our resulting algorithm are shown indicating the dependence on initial conditions.

3 Algorithm Development via the Majorize-Minimize Principle

The majorize-minimize principle seeks to replace a difficult optimization problem, in our case, non-smooth, with a simpler (smooth) one having the same solution. In the case of minimization, the idea is to find a non-unique surrogate function that majorizes the objective function of interest and then to minimize this. Here we are able to replace the non-smooth element of the objective function with a quadratic function and then iterate toward the solution.

[2] The case of $1 < q \leq 2$ is also accommodated but does not deliver the required parsimony.

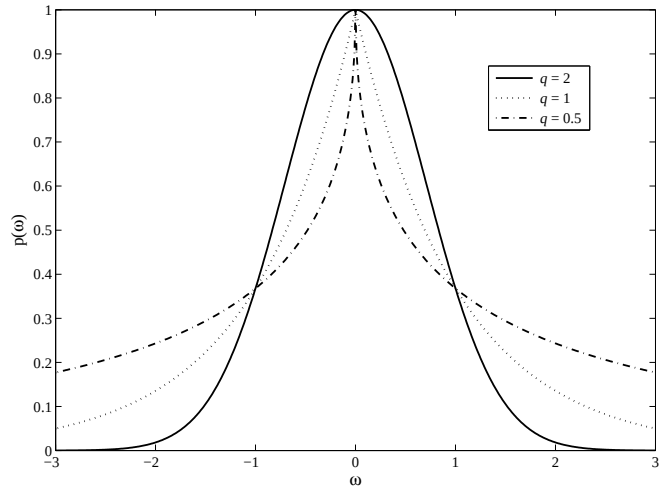

Fig. 1 Prior distribution functions associated with penalties of the form $-\|\boldsymbol{\omega}\|_q^q$ for $q \in \{\frac{1}{2}, 1, 2\}$ and scalar ω. The penalty function sharpens considerably around the origin so that small values will contribute much more strongly to the likelihood function by having their values set to zero.

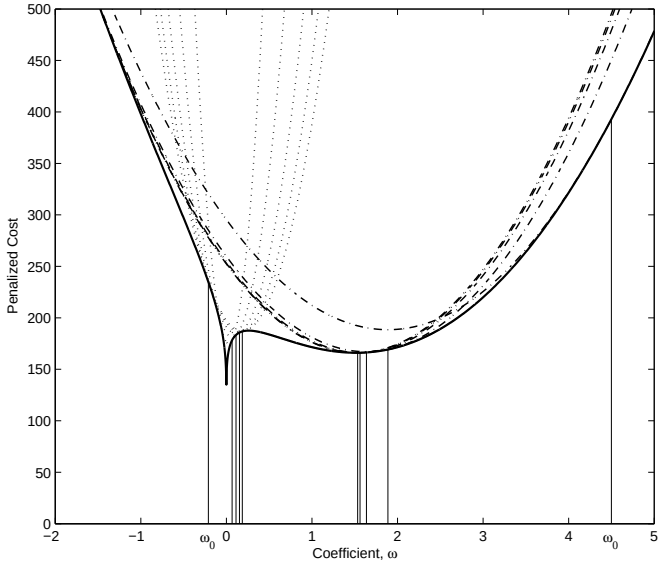

Fig. 2 The objective function for a simple, penalized linear regression with $q = 0.25$ & $\rho = 3.5$ showing the lack of convexity and dependence on initial conditions of the resulting algorithm. The leftmost sequence is shown converging to the "sharp" minimum (dominated by the penalty) while the rightmost is converging towards the broader minimum (dominated by the data misfit).

Let $\boldsymbol{\omega}(n)$ denote the value of the coefficient vector at the n^{th} step in an iterative procedure, then a function, $S\left(\boldsymbol{\omega}|\boldsymbol{\omega}(n)\right)$, is said to majorize the function, $J\left(\boldsymbol{\omega}\right)$, if it is everywhere greater than J and is tangent to it at $\boldsymbol{\omega}(n)$ (e.g. [23]), i.e.:

$$\begin{aligned} S\left(\boldsymbol{\omega}(n)|\boldsymbol{\omega}(n)\right) &= J\left(\boldsymbol{\omega}(n)\right) \\ S\left(\boldsymbol{\omega}|\boldsymbol{\omega}(n)\right) &\geq J\left(\boldsymbol{\omega}\right) \forall \boldsymbol{\omega} \end{aligned} \tag{4}$$

Majorization is closed under the operations of addition and multiplication.

Such a function that majorizes a convex objective function can itself be minimized (w.r.t. $\boldsymbol{\omega}$), often analytically, and this fact can be exploited. The majorizing function, $S(.,.)$, acts as a *surrogate* for the original objective function. The *descent* property (e.g. [23]) then guarantees that the value of $J\left(\boldsymbol{\omega}\right)$ never increases, as follows:

$$\begin{aligned} J\left(\boldsymbol{\omega}(n+1)\right) &= S\left(\boldsymbol{\omega}(n+1)|\boldsymbol{\omega}(n)\right) \\ &\quad + J\left(\boldsymbol{\omega}(n+1)\right) - S\left(\boldsymbol{\omega}(n+1)|\boldsymbol{\omega}(n)\right) \\ &\geq S\left(\boldsymbol{\omega}(n)|\boldsymbol{\omega}(n)\right) + J\left(\boldsymbol{\omega}(n)\right) - S\left(\boldsymbol{\omega}(n)|\boldsymbol{\omega}(n)\right) \\ &= J\left(\boldsymbol{\omega}(n)\right) \end{aligned}$$

owing to (4) since $S\left(\boldsymbol{\omega}(n+1)|\boldsymbol{\omega}(n)\right) \geq S\left(\boldsymbol{\omega}(n)|\boldsymbol{\omega}(n)\right)$.

We outline the derivation of a very simple algorithm of Newton-Raphson type for the penalized least-squares estimation of the kFDA coefficients (c.f. [16]). The objective function, $J\left(\boldsymbol{\omega}\right)$, is written as the sum of two functions,

$$J_e\left(\boldsymbol{\omega}\right) = \frac{1}{2}\|\hat{\boldsymbol{y}} - \tilde{\mathrm{K}}\boldsymbol{\omega}\|_2^2, J_p\left(\boldsymbol{\omega}\right) = \rho N \|\boldsymbol{\omega}\|_q^q$$

where

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \frac{N}{N_1}\mathbf{1}_{N_1} \\ -\frac{N}{N_2}\mathbf{1}_{N_2} \end{bmatrix} \in \mathbb{R}^N, \tilde{\mathrm{K}} = \begin{bmatrix} \mathbf{1}_N & K \end{bmatrix} \in \mathbb{R}^{N\times(N+1)}$$

giving:

$$J\left(\boldsymbol{\omega}\right) = \frac{1}{2}\|\hat{\boldsymbol{y}} - \tilde{\mathrm{K}}\boldsymbol{\omega}\|_2^2 + \rho N \|\boldsymbol{\omega}\|_q^q \tag{5}$$

It is clear that in the case of interest, $0 < q \leq 1$, no closed-form solution exists for the minimization of (5); however, by exploiting the fact that $|\omega|^q$ is convex on $\mathbb{R}_+$ and $|\omega|^q = \left(\omega^2\right)^{\frac{q}{2}}$ it can be shown that $J_p\left(\boldsymbol{\omega}\right)$ is majorized at every point, $\boldsymbol{\omega}(n)$, by a quadratic function thus:

$$\begin{aligned} J_p\left(\boldsymbol{\omega}\right) = \rho N \|\boldsymbol{\omega}\|_q^q &\leq \frac{\rho}{2} N \sum_{i=1}^{i=d} \left(\frac{q\omega_i^2}{|\omega_i(n)|^{2-q}} + (2-q)|\omega_i(n)|^q \right) \\ &= \frac{\rho}{2} N \left(q\boldsymbol{\omega}^{\mathsf{T}} \mathrm{B}\left(\boldsymbol{\omega}(n)\right)\boldsymbol{\omega} + (2-q)\|\boldsymbol{\omega}(n)\|_q^q \right) \end{aligned} \tag{6}$$

with $\mathrm{B}\left(\boldsymbol{\omega}(n)\right) = \mathrm{diag}\left\{|\omega_i(n)|^{q-2}\right\}$. The result arises from the relationship $g(x) \geq g(y) + dg(y)(x-y) \forall x, y$ (see e.g. [23]) and is ascribed to [24]. The function, $J\left(\boldsymbol{\omega}\right)$, in equation (5) is therefore majorized when the second term

on the RHS is replaced by the upper bound given in (6) giving a quadratic surrogate:

$$S\left(\boldsymbol{\omega}|\boldsymbol{\omega}(n)\right) = \boldsymbol{\omega}^{\mathsf{T}}\tilde{\mathrm{K}}^{\mathsf{T}}\hat{\boldsymbol{y}} - \frac{1}{2}\boldsymbol{\omega}^{\mathsf{T}}\left(\tilde{\mathrm{K}}^{\mathsf{T}}\tilde{\mathrm{K}} + \rho N q \mathrm{B}\left(\boldsymbol{\omega}(n)\right)\right)\boldsymbol{\omega}$$

(omitting constant terms in $\boldsymbol{\omega}$) that has the ascent property and which can be minimized analytically w.r.t $\boldsymbol{\omega}$. Setting the gradient of $S\left(\boldsymbol{\omega}|\boldsymbol{\omega}(n)\right)$ to zero, solving for $\boldsymbol{\omega}$ and identifying $\boldsymbol{\omega}$ with $\boldsymbol{\omega}(n+1)$ gives the following iteration

$$\boldsymbol{\omega}(n+1) = \left(\tilde{\mathrm{K}}^{\mathsf{T}}\tilde{\mathrm{K}} + \rho N q \mathrm{B}\left(\boldsymbol{\omega}(n)\right)\right)^{-1}\tilde{\mathrm{K}}^{\mathsf{T}}\hat{\boldsymbol{y}} \tag{7}$$

assuming no $\omega_i(0) = 0$.

This last condition raises a potential problem: when the elements of $\boldsymbol{\omega}(n)$ approach zero – to be expected when a sparse solution emerges – the surrogate, $S\left(\boldsymbol{\omega}|\boldsymbol{\omega}(n)\right)$, is no longer defined. The authors of [16] have shown that the addition of a small positive quantity to the denominator of the diagonal elements of $\mathrm{B}\left(\boldsymbol{\omega}(n)\right)$ retains their maximum likelihood interpretation and that this quantity can be allowed to decay to zero in the limit so that the original problem is solved. They also present a method for the informed selection of its value.

To avoid the difficulty we take the pragmatic approach advocated in [25–27] by re-writing $\mathrm{B}\left(\boldsymbol{\omega}(n)\right) = \Psi_n^{-2}$ with $\Psi_n = \mathrm{diag}\left\{|\omega_i(n)|^{\frac{2-q}{2}}\right\}$ leading to

$$\boldsymbol{\omega}(n+1) = \Psi_n\left(\Psi_n\tilde{\mathrm{K}}^{\mathsf{T}}\tilde{\mathrm{K}}\Psi_n + \rho N q \mathrm{I}_{N+1}\right)^{-1}\Psi_n\tilde{\mathrm{K}}^{\mathsf{T}}\hat{\boldsymbol{y}} \tag{8}$$

Evidently, if any $\omega_i(0) = 0$ this results in the permanent exclusion of the i^{th} coefficient from the optimization so we set $\boldsymbol{\omega}\left(0\right) \neq \mathbf{0}$. Furthermore, during iteration one or more ω_i may head towards zero which could affect the convergence toward the minimizer of the objective function. Our own experience here, in [28–30] and that of [25–27] suggests no practical problem. Indeed [27] shows for the penalized least-squares case with convex objective function that if $\boldsymbol{\omega}\left(0\right) \neq \mathbf{0}$ then, with probability one, no ω_i achieves zero in a finite number of iterations so from a practical viewpoint there should be no difficulty. Here, convergence is declared when the relative change in the objective function is less than some threshold, $\epsilon \ll 1$ (here $\epsilon = 10^{-5}$). We denote the resulting classifiers kFDA_q.

Majorize-minimize algorithms display, typically, a linear rate of convergence in the vicinity of the optimum in contrast to the quadratic convergence of a typical Newton-Raphson approach. On the plus side, though, they tend to require simpler computations at each step and it is possible, therefore, for the approach to be faster in clock-time. Computational speed-ups such as Schultz-Hotelling acceleration and successive over-relaxation have been suggested to improve matters [23] but these issues are not addressed here. The iteration (8) has the same computational overhead as the familiar iteratively re-weighted least-squares algorithm used widely in generalized linear modelling.

4 Performance Comparison with Previous Methods

To evaluate the performance of kFDA$_q$ extensive experimentation has been carried out on the well-studied 13 datasets of the FIRST IDA repository [31] that have been used to benchmark a number of machine learning techniques, see e.g. [14,4,32] among 24 recent articles revealed by a citation search of machine learning publications to reveal the current leaders in these samples. An additional seven sets selected from the UCI repository [33] on the basis of having no fewer than 100 samples, no more than 70% majority class and less than 10% missing data were also studied[3]. These have also been widely studied and a citation search disclosed a further 14 articles. The methodology outlined in [14] was followed throughout, enabling direct comparison with a number of techniques. In particular, we have made a detailed comparison with the results from [14] because that work is specifically concerned with complexity control and proved, at its time of writing, to be a consistently high performer in terms of both accuracy and sparsity. The results in [14] are computed using a decision threshold given by $0.5N(1/N_1 - 1/N_2)$ – the mid-point between target values and since our main performance comparison is with this work, we too have adopted it while recognizing that it will not necessarily be optimal for any given sample.

We have updated the comparison with the results of the current best performers in all 20 domains[4]. All experiments are carried out using the Matlab environment [34] and the Gaussian Radial Basis Function kernel, $k(\boldsymbol{u}, \boldsymbol{v}) = \exp\left(-\frac{\|\boldsymbol{u}-\boldsymbol{v}\|^2}{2\sigma^2}\right)$.

The experimental method is based on 100 random partitions of the samples[5]. To select regularization and kernel parameters, five-fold cross-validation is used on each of the first five realizations and those corresponding to the realization with lowest misclassification rate are used to test the remaining realizations. Here we examine two situations: selection of classifiers (i) for minimum misclassification rate (MCR) and (ii) for minimum number of retained samples (NRS). Each is then applied to all test partitions. In the literature most authors have reported results to one decimal place so we too have followed that convention.

4.1 FIRST IDA Datasets

Table 1 shows percentage mean MCR and NRS calculated for the test sets for each of the 13 domains. We report kFDA$_q$ for $q \in \{1, 0.5\}$, the better of the two methods proposed in [14] referred to generically as kFDA$_{\text{OLS}}$, and the current best results from our citation search. From Table 1 it can be seen that the previously published best MCR remains best in 3/13 cases

[3] Missing data were simply excised from the samples.

[4] It should be noted that the methodology of many of the techniques used in the comparisons may not conform to that of [14].

[5] Twenty partitions for "Splice" and "Image" datasets.

("Heart", "Twonorm" and "Waveform") and is joint best in "S. Flare". The kFDA_1 classifier (selected for minimum MCR) proves most accurate in 6/13 domains and most accurate on average across all domains. Its sparseness is, however, relatively poor in all 13 cases when compared to kFDA_{OLS} but improves on the other published results in 2/4 cases where a sparse solution is reported[6]. Choosing the kFDA_1 classifier for minimum NRS instead, improves sparsity overall but results in a small loss of accuracy. In particular kFDA_{OLS} proves sparser in 12/13 cases but the situation is reversed (10/13 with equality in "Image") when considering accuracy.

To encourage further sparseness, q is reduced to 0.5. Selecting for minimum MCR, $\text{kFDA}_{0.5}$ exhibits highest accuracy in 6/13 cases (equal to kFDA_1 in "Diabetes" and "Titanic", and to the best published in "S. Flare") while exceeding or matching the sparseness of kFDA_{OLS} in 9/13 cases. Selecting $\text{kFDA}_{0.5}$ for minimum NRS even simpler models are frequently found (11/13 are most sparse across all methods with lowest average NRS) but again with a small loss of accuracy. The loss of convexity in the objective function does not appear to have led to any major loss in performance here.

4.2 UCI Datasets

Table 2 presents results from the second set of benchmarks[7] The overall picture here is much the same as for the FIRST IDA data with the previously published best MCR remaining best in only 2/7 cases. kFDA_1 is most accurate in 4/7 cases and likewise on average. It is interesting to note the rather spectacular accuracy achieved for the "Credit" dataset by the method reported in [35]. This involves an initial clustering stage followed by application of a neural network. The clustering stage is used to identify and remove isolated and inconsistent clusters from the training set. Such an approach is bordering on the use of prior knowledge rather than a direct (agnostic) learning approach and this might explain the uncharacteristic accuracy. The MCR value calculated from [35] that corresponds to the agnostic situation is $10.6 \pm 1.4\%$ which is more in line with the other results.

Again selecting to reduce NRS has the desired effect but at the cost of reducing accuracy. Reducing q to 0.5 and selecting for minimum MCR induces further sparsity improving mean sparsity considerably but with a small degradation in accuracy. Selecting for minimum NRS maximizes sparsity without too much loss of accuracy but still does not match the sparsity of kFDA_{OLS}. It should be noted that none of the previously reported

[6] In 9/13 cases NRS is shown as 100% because the reported method, although a kernel machine and therefore a candidate for sparsity control, no attempt has been made to achieve this. Inclusion of these values has a strong effect on the average NRS value in this column.

[7] The results for kFDA_{OLS} have not been previously reported but have been computed by the authors.

Table 1 FIRST IDA Repository Database: Comparison of mean misclassification rate and sparsity for the proposed algorithm, kFDA_q, kFDA_{OLS} [14], and the best published algorithm: Import Vector Machine [36] (★), conventional kFDA (△) [7], Reformative kFDA (▲) [7], Naive Kernel-based Nonlinear Method [8] (▽), Fast Kernel-based Nonlinear Method [8] (▼), Kernel Logistic Regression [37] (□), Sparse kFDA with Linear Loss [38] (■), Linear Programming AdaBoost [39] (◊), and Suppressed Kernel Sample Space Projection [40] (⧫). Bold type – best performance/sparsity, italic type – sample size.

Database	Published	kFDA_{OLS}	kFDA_1		$\text{kFDA}_{0.5}$	
	Best (%)	(%)	MCR (%)	NRS (%)	MCR (%)	NRS (%)
Banana	10.3 ± 0.5	10.7 ± 0.5	9.8 ± 0.1	12.8 ± 0.1	**9.6 ± 0.1**	**9.6 ± 0.1**
400	5.3 ± 1.8★	7.3	15.3	10.5	**4.5**	**4.5**
B.Cancer	22.7 ± 4.4	25.3 ± 4.1	21.3 ± 3.7	22.8 ± 4.4	**20.5 ± 4.0**	25.4 ± 4.0
200	100.0 △	3.5	12.5	6.5	3.5	**1.0**
Diabetes	22.1 ± 1.9	23.1 ± 1.8	**21.6 ± 1.4**	22.8 ± 1.6	**21.6 ± 1.6**	**21.6 ± 1.6**
468	100.0 △	2.1	4.1	2.4	**1.3**	**1.3**
German	21.3 ± 2.1	24.0 ± 2.1	**21.0 ± 1.8**	23.5 ± 2.0	23.5 ± 2.0	23.5 ± 2.0
700	100.0 △	**1.1**	9.4	2.7	**1.1**	**1.1**
Heart	**10.8 ± 2.6**	15.8 ± 3.4	14.6 ± 3.0	14.6 ± 3.0	14.8 ± 3.3	16.0 ± 3.2
170	16.0 ▲	**1.7**	10.6	10.6	2.4	1.8
Image	1.8 ± N/A	2.8 ± 0.6	**1.6 ± 0.5**	2.8 ± 0.6	2.1 ± 0.3	2.1 ± 0.3
1300	100.0 □	21.5	24.6	23.5	**15.5**	**15.5**
Ringnorm	1.5 ± 0.1	1.6 ± 0.1	**1.4 ± 0.0**	**1.4 ± 0.0**	1.5 ± 0.0	1.8 ± 0.0
400	6.0 ■	1.8	5.5	5.5	2.8	**0.8**
S.Flare	**31.6 ± 1.9**	33.5 ± 1.6	33.0 ± 1.7	33.0 ± 1.7	**31.6 ± 1.9**	32.2 ± 1.8
666	100.0 ▽	1.4	27.9	27.9	2.4	**0.5**
Splice	9.3 ± 0.7	11.7 ± 0.6	7.1 ± 0.7	7.9 ± 0.9	**7.0 ± 0.8**	7.8 ± 0.8
1000	100.0 ◊	**33.0**	85.0	71.9	75.3	54.3
Thyroid	1.4 ± 0.9	4.5 ± 2.4	**1.0 ± 0.9**	2.3 ± 1.4	1.1 ± 0.9	2.3 ± 1.3
140	16.4 ▼	16.4	22.9	10.7	12.9	**2.1**
Titanic	21.7 ± 0.3	22.4 ± 1.0	**21.1 ± 0.2**	22.1 ± 0.2	**21.1 ± 0.2**	22.7 ± 0.3
150	100.0 ▽	7.3	64.7	28.0	4.7	**1.3**
Twonorm	**2.3 ± 0.1**	2.7 ± 0.2	2.4 ± 0.0	2.4 ± 0.0	2.6 ± 0.0	2.6 ± 0.0
400	100.0 ⧫	2.5	7.0	7.0	**1.3**	**1.3**
Waveform	**9.3 ± 0.4**	10.0 ± 0.4	9.4 ± 0.1	10.2 ± 0.1	10.0 ± 0.1	10.7 ± 0.1
400	100.0 ◊	3.5	6.8	5.5	3.0	**2.3**
Mean	12.8	14.5	**12.7**	13.7	12.8	13.7
	72.6	7.9	22.8	16.4	10.0	**6.7**

best performers on the UCI Dataset can be assessed for sparsity in the sense meant here.

To summarize, Table 3 presents the average results across all 20 datasets and demonstrates the trade-off between accuracy and sparsity in these methods. In particular, it can be seen that the proposed method can produce parsimonious solutions with an accuracy comparable to the best published methods. While differences in results are not great in most cases, especially when their spread is taken into account, it is fair to say that kFDA_q offers convincingly competitive performance across a range of classification tasks.

5 Conclusion

We have introduced an algorithm for the parsimonious solution of the binary kFDA problem through the application of the majorize-minimize principle.

Table 2 UCI Repository Database: Comparison of mean misclassification rate and sparsity for the proposed algorithm, kFDA_q, kFDA_{OLS}, and the best published algorithm: Clustering + Neural Network [35] (★), C4.5 Decision Tree Learning [41] (△), Heteroscedastic LDA + Support Vector Machine [42] (▲), Multi Feature Subsets + C4.5 Decision Tree Learning [43] (▽), Local Boosted Discriminant Projections + Support Vector Machine [42] (▼), Naive Baysian [44] (□), Neural Network [45] (■). Bold type – best performance/sparsity, italic type – sample size.

Database	Published	kFDA_{OLS}	kFDA_1		$\text{kFDA}_{0.5}$	
	Best (%)	(%)	MCR (%)	NRS (%)	MCR (%)	NRS (%)
Credit	**2.0 ± 0.7**	13.3 ± 1.4	10.8 ± 1.4	14.5 ± 1.4	13.0 ± 1.4	13.3 ± 1.4
327	N/A ★	**2.1**	27.2	15.6	6.4	3.4
Chess	0.8 ± N/A	1.8 ± 0.2	**0.7 ± 0.2**	0.9 ± 0.2	**0.7 ± 0.1**	0.9 ± 0.2
1598	N/A △	**7.8**	44.7	34.0	20.0	16.1
Ionosphere	4.9 ± N/A	2.8 ± 0.8	2.8 ± 0.8	3.6 ± 1.0	**2.4 ± 0.8**	3.1 ± 0.9
176	N/A ▲	10.2	26.7	24.4	9.7	**5.1**
Liver	**24.3 ± N/A**	25.9 ± 2.0	25.1 ± 2.1	29.4 ± 2.3	26.6 ± 2.3	28.5 ± 2.2
173	N/A ▽	6.4	8.7	**4.6**	5.2	**4.6**
Sonar	10.1 ± N/A	6.6 ± 1.9	**4.4 ± 1.5**	9.5 ± 2.2	6.8 ± 1.9	6.8 ± 1.9
104	N/A ▼	**45.2**	94.2	83.7	51.0	51.0
WBC	2.6 ± N/A	2.8 ± 0.8	**2.1 ± 0.5**	**2.1 ± 0.5**	2.9 ± 0.7	2.9 ± 0.7
342	N/A □	4.4	2.3	2.3	**0.6**	**0.6**
WDBC	2.5 ± 2.1	2.6 ± 0.6	**1.1 ± 0.4**	2.2 ± 0.6	1.8 ± 0.6	2.6 ± 0.7
285	N/A ■	4.9	44.6	14.0	7.0	**4.2**
Mean	6.7	8.0	**6.7**	8.9	7.8	8.3
	N/A	**11.6**	35.5	25.5	14.3	12.1

Table 3 Average misclassification rates (MCR) and numbers of samples retained (NRS) across all data sets.

	Published	kFDA_{OLS}	kFDA_1		$\text{kFDA}_{0.5}$	
	Best (%)	(%)	MCR (%)	NS (%)	MCR (%)	NS (%)
MCR	10.7	12.2	**10.6**	12.1	11.1	11.8
NRS	N/A	9.2	26.9	18.9	11.6	**8.8**

The method exploits the correspondence between FDA and least-squares to pose a penalized least-squares problem that is known to have sparsity-inducing properties. Flexibility is provided by means of the so-called "kernel trick".

The resulting formulation leads to a non-smooth optimization problem of the same "size" as the (training) sample. The majorize-minimize principle uses a quadratic (hence smooth) upper bound on the objective function that permits a step-wise descent towards a minimum – global or local depending on the choice of norm in the penalty function. While convergence to the minimum cannot be proven, theoretical and empirical justification is given as to why the method succeeds. The optimization is easily solved iteratively with the same computational overhead as the widely-used iteratively re-weighted least squares algorithm.

Extensive comparisons have been carried out across two well-studied and substantial benchmark datasets using 100 random data partitions and five-fold cross-validation for parameter selection. The outcome demonstrates

that the proposed method delivers results as accurate and/or parsimonious, or better, than a number of leading machine learning algorithms. Selective choice of penalizing norm function has been shown to aid parsimony with little or no degradation in performance, leading to highly compact, accurate dichotomizers.

Extension of the method to the multinomial situation has so far eluded us because the solution delivers as many operators as there are classes and it is unclear how these can be employed to deliver sparse, multinomial kFDA – this problem is currently under examination.

6 Originality and Contribution

The problem of binary classification is still an important one despite many advances over the last three-quarters of a century. Extensions to Fisher's original linear discriminant analysis over that time have recently arrived at the stage of arbitrary non-linear mappings though the application of the so-called "kernel trick". While having manifest advantages, this leads to a formulation that is dominated by the size of the training sample which is usually unnecessarily complex and requires regularization to avoid overspecialization. The present work addresses these problems simultaneously through the use of a non-smooth regularizer and the application of a "majorize-minimize" algorithm to overcome the difficulties presented by the lack of smoothness in achieving optimality. It does this via the connection of FDA with least-squares to yield a very simple iterative algorithm. The use of this approach in connection with the Fisher problem is, to the best of our knowledge, novel.

The resulting algorithm has the same computational overhead as iteratively re-weighted least-squares and, while convergence has not been proved, its use is theoretically justified. Comprehensive comparisons across 20 publicly available machine learning benchmarks reveal that the method provides a classification performance that is comparable with or better than many state-of-the-art methods. The extensive nature of these comparisons themselves form a point of comparison for future developments in the area.

References

1. Duda, R., Hart, P., and Stork, D. (2001) *Pattern Classification*. John Wiley & Sons, New York, NY.
2. Schölkopf, B. and Smola, A. (2002) *Learning with Kernels*. MIT Press, Cambridge, MA.
3. Baudat, G. and Anouar, F. (2000) Generalized discriminant analysis using a kernel approach. *Neur. Comput.*, **12**: 2385–2404.
4. Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Smola, A., and Müller, K. (2003) Constructing descriptive and discriminative nonlinear features: Rayleigh Coefficients in kernel feature spaces. *IEEE T. Patt. Anal.*, **25**: 623–628.

5. Roth, V. and Steinhage, V. (1999) Nonlinear discriminant analysis using kernel functions. In Solla, S., Leen, T., and Müller, K. (eds.), *Advances in neural information processing systems*, vol. 12, pp. 568–574.
6. Xu, Y., Yang, J., Lu, J., and Yu, D. (2004) An efficient renovation on kernel Fisher discriminant analysis and face recognition experiments. *Patt. Recogn.*, **37**: 2091–2094.
7. Xu, Y., Yang, J., and Yang, J. (2004) A reformative kernel Fisher discriminant analysis. *Patt. Recogn.*, **37**: 1299–1302.
8. Xu, Y., Zhang, D., Jin, Z., Li, M., and Yang, J. (2006) A fast kernel-based nonlinear discriminant analysis for multi-class problems. *Patt. Recogn.*, **39**: 1026–1033.
9. Yang, J., Frangi, A., Yang, J., Zhang, D., and Jin, Z. (2005) KPCA plus LDA: A complete kernel fisher discriminant framework for feature extraction and recognition. *IEEE T. Patt. Anal.*, **27**: 230–244.
10. Liang, Z. and Shi, P. (2004) An effiecient and effective method to solve kernel Fisher discriminant analysis. *Neurocomp.*, **61**: 485–493.
11. Liang, Z. and Shi, P. (2005) Kernel direct discriminant analysis and its theoretical foundation. *Patt. Recogn.*, **38**: 445–447.
12. Lu, J., Plataniotis, K. N., and Venetsanopoulos, A. N. (2003) Face recognition using kernel direct discriminant analysis algorithms. *IEEE T. Neur. Networ.*, **14**: 117–126.
13. Andelić, E., Schafföner, M., Katz, M., Krüger, S., and Wendemuth, A. (2006) Kernel least-squares models using updates of the pseudoinverse. *Neurocomp.*, **18**: 2928–2935.
14. Billings, S. and Lee, K. (2002) Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm. *Neur. Networ.*, **15**: 263–270.
15. Abe, S. (2006) Sparse least-squares support vector training in the reduced empirical feature space. *Patt. Anal. Applic.*, **10**: 203–214.
16. Hunter, D. and Li, R. (2005) Variable selection using MM algorithms. *Ann. Stat.*, **33**: 1617–1642.
17. Fukunaga, K. (1990) *Introduction to Statistical Pattern Recognition*. Academic Press, London.
18. Hastie, T., Tibshirani, R., and Friedman, J. (2001) *The Elements of Statistical Learning*. Springer Series in Statistics, Springer-Verlag, New York, NY.
19. Bishop, C. M. (2006) *Pattern Recognition and Machine Learning*. Springer, New York.
20. Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004) Least angle regression. *Ann. Stat.*, **32**: 407–499.
21. Osborne, M., Presnell, B., and Turlach, B. (2000) On the LASSO and its dual. *J. Comput. Graph. Stat.*, **9**: 319–337.
22. Roth, V. (2004) The generalised LASSO. *IEEE T. Neur. Networ.*, **15**: 16–28.
23. Lange, K., Hunter, D., and Yang, I. (2000) Optimization transfer using surrogate objective functions. *J. Comput. Graph. Stat.*, **9**: 1–59.
24. Dutter, R. and Huber, P. (1981) Numerical methods for the non-linear robust regression problem. *J. Stat. Comput. Sim.*, **13**: 79–113.
25. Krishnapuram, B., Carin, L., Figueiredo, M., and Hartemeink, A. (2005) Sparse multinomial logistic regression: fast algorithms and generalization bounds. *IEEE T. Patt. Anal.*, **27**: 957–968.
26. Figueiredo, M. (2003) Adaptive sparseness for supervised learning. *IEEE T. Patt. Anal.*, **25**: 1150–1159.

27. Figueiredo, M., Bioucas-Dias, J., and Nowak, R. (2007) Majorization-minimization algorithms for wavelet-based image restoration. *IEEE T. Image Process.*, **16**: 2980–2991.
28. Pasupa, K., Harrison, R., and Willett, P. (2007) Parsimonious kernel Fisher discrimination. In Martí, J., Benedí, J., Mendonça, A., and Serrat, J. (eds.), *Pattern Recognition and Image Analysis*, vol. 4477 of *Lecture Notes in Computer Science*, pp. 531–538, Springer.
29. Harrison, R. and Dodd, T. (2007) Estimation of parsimonious discrete Volterra series. In Xia, X. and Allgower, F. (eds.), *Proceedings of 7th IFAC Symposium on Nonlinear Control Systems*, Pretoria, Elsevier, in press.
30. Harrison, R. and Kennedy, R. (2008) Automatic covariate selection in logistic models for chest pain diagnosis: a new approach. *Comput. Prog. Meth. Biomed.*, **89**: 301–312.
31. Rätsch, G. (2001), FIRST IDA Benchmark Repository. World Wide Web, `http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm`.
32. Rätsch, G., Onoda, T., and Müller, K. (2001) Soft margins for AdaBoost. *Mach. Learn.*, **42**: 287–320.
33. Newman, D., Hettich, S., Blake, C., and Merz, C. (1998), UCI Repository of machine learning databases. World Wide Web, `http://www.ics.uci.edu/~mlearn/MLRepository.html`.
34. MathWorks (2005) *Matlab version 7.3*. The MathWorks Inc., Natick, MA.
35. Hsieh, N. (2005) Hybrid mining approach in the design of credit scoring models. *Expert Syst. Appl.*, **28**: 655–665.
36. Zhu, J. and Hastie, T. (2005) Kernel logistic regression and the import vector machine. *J. Comput. Graph. Stat.*, **14**: 185–205.
37. Keerthi, S., Duan, K., Shevade, S., and Poo, A. (2005) A fast dual algorithm for kernel logistic regression. *Mach. Learn.*, **61**: 151–165.
38. Mika, S., Rätsch, G., and Müller, K. (2001) A mathematical programming approach to the kernel Fisher algorithm. In Leen, T., Dietterich, T., and Tresp, V. (eds.), *Advances in Neural Information Processing Systems*, London, England, vol. 13, pp. 591–597, MIT Press.
39. Sun, Y., Todorovic, S., and Li, J. (2007) Increasing the robustness of boosting algorithms within the linear-programming framework. *J. VLSI Signal Proc.*, **48**: 5–20.
40. Washizawa, Y. and Yamashita, Y. (2006) Kernel projection classifiers with suppressing features of other classes. *Neural Comput.*, **18**: 1932–1950.
41. Last, M. and Maimon, O. (2004) A compact and accurate model for classification. *IEEE T. Knowl. Data Eng.*, **16**: 203–215.
42. Masip, D., Kuncheva, L., and Vitrià, J. (2005) An ensemble-based method for linear feature extraction for two-class problems. *Patt. Anal. Appl.*, **8**: 227–237.
43. Latinne, P., Debeir, O., and Decaestecker, C. (2000) Different ways of weakening decision trees and their impact on classification accuracy of DT combination. In Kittler, J. and Roli, F. (eds.), *Multiple Classifier Systems*, Cagliari, vol. 1857 of *Lecture Notes in Computer Science*, pp. 200–209, Springer.
44. Chien, B., Lin, J., and Yang, W. (2006) A classification tree based on discriminant functions. *J. Inf. Sci. Eng.*, **22**: 573–594.
45. Park, Y., Kim, B., and Chun, S. (2006) New knowledge extraction technique using probability for case-based reasoning: application to medical diagnosis. *Expert Syst.*, **23**: 2–20.